PEWO: a collection of workflows to benchmark phylogenetic placement

Benjamin Linard [1,2,*], Nikolai Romashchenko [1], Fabio Pardi [1] and Eric Rivals [1,3*]

[1]LIRMM, University of Montpellier, CNRS, Montpellier, France
[2]SPYGEN, 17 Rue du Lac Saint-André, 73370 Le Bourget-du-Lac, France
[3]Institut Français de Bioinformatique, CNRS UMS 3601, Évry, France.

*To whom correspondence should be addressed.

Abstract

Motivation: Phylogenetic placement (PP) is a process of taxonomic identification for which several tools are now available. However, it remains difficult to assess which tool is more adapted to particular genomic data or a particular reference taxonomy. We developed PEWO, the first benchmarking tool dedicated to PP assessment. Its automated workflows can evaluate PP at many levels, from parameter optimisation for a particular tool, to the selection of the most appropriate genetic marker when PP-based species identifications are targeted. Our goal is that PEWO will become a community effort and a standard supportred for future developments and applications of PP.
Availability: `https://github.com/phylo42/PEWO`
Contact: benjamin.linard@lirmm.fr; rivals@lirmm.fr
Supplementary : Supplementary data is available at page 4.

1 Introduction

When a reference phylogeny is available, taxonomic identification of biological sequences can be achieved with phylogenetic placement (PP). PP provides the most informative type of classification because each query sequence is assigned to its putative origin in the tree. PP can be applied in many contexts, including community ecology, species diversity, or medical studies. Several PP tools were developed for these purposes (Matsen *et al.*, 2010; Berger *et al.*, 2011; Mirarab *et al.*, 2012; Zheng *et al.*, 2018), with four recent tools capable of processing larger sequence volumes (Barbera *et al.*, 2018; Linard *et al.*, 2019; Czech and Stamatakis, 2019; Balaban *et al.*, 2020). In the preliminary phase of experimental design, assessing which tools answer the needs of a given application remains a tedious task often involving manual tests (Mangul *et al.*, 2019). Strikingly, PP has a broad range of applications, but lacks user guidelines and benchmarking. Some procedures to evaluate PP accuracy were proposed (Matsen *et al.*, 2010), but never automated via a dedicated software. Benchmarking is essential to determine which tool suits better a given metagenomic task or a specific dataset (Sczyrba *et al.*, 2017).

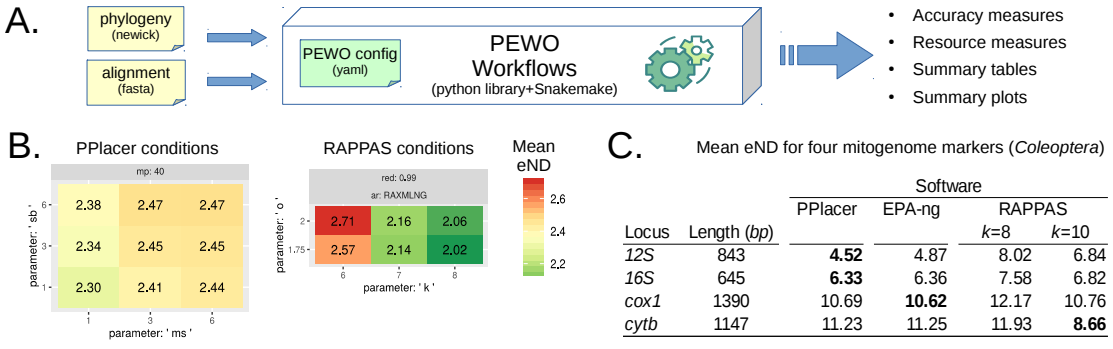
To fill this gap, we developed PEWO (Placement Evaluation WOrkflows), the first tool dedicated to PP benchmarking. PEWO automatizes evaluation procedures (which were not implemented for the community), and introduces novel procedures. Beyond benchmarking, PEWO can help decision-making in any metagenomic or metabarcoding project for PP-based taxonomic identification. With applications ranging from parameter optimization on particular genomic data, to the selection of the most appropriate genetic marker, PEWO provides the user community with standardized workflows for easy and reproducible assessment of PP analyses.

2 Overview

PEWO implements evaluation workflows in Python and Snakemake (Köster and Rahmann, 2012), whose framework ensures flexibility, platform independence, and reproducibility. Each workflow automatically performs multiple steps from query generation up to summary plots/tables, and can be tailored via Snakemake configuration files. PEWO and its dependencies are easily installed via a conda virtual environment. Currently, PEWO incorporates five state-of-the-art PP tools, which cover a majority of PP uses: EPA(RAxML), PPlacer, EPA-ng, RAPPAS and APPLES. Four are alignment-based tools, while RAPPAS is alignment-free. As input, each workflow takes a phylogenetic tree and the reference multiple sequence alignment from which it was built (Figure 1). Optionally, the user can provide a set of query sequences. Below we describe the workflows and some of their applications.

2.1 PEWO procedures

- *Pruning-based accuracy evaluation* (PAC): in this standard procedure for assessing placement accuracy (Matsen *et al.*, 2010; Berger *et al.*, 2011), a subset of sequences is randomly pruned from the reference phylogeny and alignment. Each pruned sequence then serves to generate queries for placement, and the accuracy of each tool is measured in number of nodes separating predicted from true placement. PEWO offers two versions of this topological metric: *Node Distance* (ND) and *expected Node Distance* (eND). The eND accounts for placement uncertainty (e.g. *likelihood weight ratios*). All selected tools are compared for a user-selected combination of parameters.
- *Likelihood-based accuracy evaluation* (LAC) is a new, faster evaluation procedure introduced in PEWO to assess *relative* accuracy of PP. It iterates the following process for a set of queries: place the query, extend the phylogeny to include that query, optimize the branch lengths of this extended tree, and return its log-likelihood (LL). The user can then compare the LL values obtained with different tools, or different settings of a same tool (e.g. by inspecting the distribution of



		Software			
		PPlacer	EPA-ng	RAPPAS	
Locus	Length (*bp*)			*k*=8	*k*=10
12S	843	**4.52**	4.87	8.02	6.84
16S	645	**6.33**	6.36	7.58	6.82
cox1	1390	10.69	**10.62**	12.17	10.76
cytb	1147	11.23	11.25	11.93	**8.66**

Fig. 1. A. Overview of PEWO inputs and outputs. B. An example of plots dynamically-generated by the PAC (Pruning-based Accuracy Evaluation) procedure on a 16S rRNA bacterial reference. Measured Mean expected Node Distances (eND) are reported (lower value = better accuracy). Panels report selected conditions for PPlacer and RAPPAS, e.g. different parameter values tested in different rows and columns. For PPlacer, varying parameters are `ms` (max-strikes, X axis) and `sb` (strike-box, Y axis). Parameter `mp` (max-pitches, grey box) is fixed. For RAPPAS, varying parameters are `k` (phylo-kmer size) and `o` (omega threshold). Parameters `red` (alignment reduction) and `ar` (software used for ancestral reconstruction) are fixed. C. Four PAC procedures were run for different Coleopteran mitogenome loci (rows) and compiled. Average expected Node Distance (eND) is measured for three tools (columns) using default parameters. For each locus, the lowest average eND is highlighted in bold. For RAPPAS, the last column shows that accuracy can be improved when increasing k-mer size (default is `k`=8). Examples B. and C. are more extensively discussed in Supplementary Materials.

the differences between LL values obtained with two different tools). See the Supplementary Materials for a more detailed description.

- *Resource evaluation (RES)*: outputs the runtime and memory usage of selected tools, with details for each placement step (e.g., profile alignment, database construction, placement...). One can compare the impact on time and memory for tool-specific parameter combinations, while searching for an appropriate accuracy/resource trade-off, or evaluate the tools' scalability with respect to input size.

2.2 Applications

PEWO procedures cover numerous use cases arising with PP, as illustrated by six exemplar applications provided on GitHub (two are reported in Figure 1B-C). As new PP tools can be incorporated in PEWO, PEWO procedures enable comparing existing and future tools on resource usage, scalability, or accuracy in a reproducible way. With PEWO, users can optimize their PP pipeline design. For instance, for a given reference (tree and alignment), determine which tool and parameter combination will maximize placement accuracy, and at which computational cost. PEWO facilitates such tests, as in Figure 1-B, which shows two plots automatically generated by the PAC procedure running PPlacer and RAPPAS for 9 and 6 parameter combinations, respectively.

As a second example, we show how PEWO can be used to compare different genetic markers available for the same taxa, as the choice of the marker may impact the accuracy of placement. For example, we evaluated the placements for four loci (*16S*, *12S*, *cox1*, *cyt*) on their associated phylogeny for 900 Coleopteran mitochondrial genomes (Linard *et al.*, 2018). Figure 1-C displays the results (reproducible via GitHub example 4) highlighting that: i) *12S* yields the most accurate placements, despite being the second shortest locus, ii) the tool achieving the best accuracy depends on the marker, and iii) with RAPPAS, a longer k-mer size is required to obtain accuracy similar or better than alignment-based methods.

2.3 Availability and implementation

PEWO, with full documentation and example workflows, is freely available from its repository URL: https://github.com/phylo42/PEWO. Its modular, well-documented, and evolvable source code enables the community to easily extend it by adding new tools, procedures, or metrics. Notably, users can develop their own evaluation procedures starting from PEWO Snakemake rules as templates for their own workflows. Any PP tool can be integrated as long as it outputs results in jplace format (a json specification, standard in PP, see (Matsen *et al.*, 2012)), can be parameterized via the command line, and is available on a conda or pip repository (see the documentation for guidelines).

3 Conclusion

Reproducibility of computational analyses in life sciences is a crucial issue, even more when large scale data comes into play, as in the case of metagenomics. With PEWO, we provide a resource that facilitates the evaluation and comparison of PP tools under a unified framework. It allies flexibility, extensibility, with ease of use, while it inherits a standardized installation procedure from the conda framework. The set of workflows in PEWO aims to grow as a community effort, and extensions are welcome. In PEWO, we introduce a likelihood-based accuracy evaluation procedure, which is complementary to existing procedures (Matsen *et al.*, 2010). PEWO will help the community in its efforts to develop future PP tools and will facilitate experimental decisions when PP is chosen as a means to species identification. With the help of future contributors, we hope that PEWO will evolve as a standard for PP benchmarking, and answer forthcoming unforeseen yet auspicious applications.

Acknowledgements

We thank Vincent Lefort for technical assistance, the ATGC bioinformatic platform, the Institut Français de Bioinformatique [ANR-11-INBS-0013].

Funding

This work has been supported by France Génomique [ANR-10-INBS-0009], MNERT fellowship to NR.

References

Balaban, M. *et al.* (2020). Apples: scalable distance-based phylogenetic placement with or without alignments. *Systematic Biology*, **69**(3), 566–578.

Barbera, P. *et al.* (2018). EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences. *Systematic Biology*, **68**(2), 365–369.

Berger, S. A. *et al.* (2011). Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. *Systematic Biology*, **60**(3), 291–302.

Czech, L. and Stamatakis, A. (2019). Scalable methods for analyzing and visualizing phylogenetic placement of metagenomic samples. *PLOS ONE*, **14**(5), e0217050.

Köster, J. and Rahmann, S. (2012). Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, **28**(19), 2520–2522.

Linard, B. *et al.* (2018). The contribution of mitochondrial metagenomics to large-scale data mining and phylogenetic analysis of coleoptera. *Molecular Phylogenetics and Evolution*, **128**, 1 – 11.

Linard, B. *et al.* (2019). Rapid alignment-free phylogenetic identification of metagenomic sequences. *Bioinformatics*, **35**(18), 3303–3312.

Mangul, S. *et al.* (2019). Systematic benchmarking of omics computational tools. *Nature Communications*, **10**(1).

Matsen, F. A. *et al.* (2010). pplacer: Linear time maximum-likelihood and bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, **11**(1), 538.

Matsen, F. A. *et al.* (2012). A format for phylogenetic placements. *PLoS ONE*, **7**(2), e31009.

Mirarab, S. *et al.* (2012). SEPP: sate -enabled phylogenetic placement. In R. B. Altman, A. K. Dunker, L. Hunter, T. Murray, and T. E. Klein, editors, *Biocomputing 2012: Proceedings of the Pacific Symposium, Kohala Coast, Hawaii, USA, January 3-7, 2012*, pages 247–258. World Scientific Publishing.

Sczyrba, A. *et al.* (2017). Critical assessment of metagenome interpretation - a benchmark of metagenomics software. *Nature Methods*, **14**(11), 1063–1071.

Zheng, Q. *et al.* (2018). HmmUFOtu: An Hmm and Phylogenetic Placement Based Ultra-Fast Taxonomic Assignment and Otu Picking Tool for Microbiome Amplicon Sequencing Studies. *Genome Biology*, **19**(1), 82.

Supplementary Material to "PEWO: a collection of workflows to benchmark phylogenetic placement"

Authors: B. Linard, N. Romashchenko, F. Pardi, E. Rivals.

Likelihood-based accuracy evaluation (LAC)

This procedure is another way of testing and comparing phylogenetic placement tools. Each tested reference dataset consists of a reference alignment (*refA*), a reference tree (*refT*), and a dataset of query sequences (*QS*). In the PEWO LAC procedure, the following steps are repeated for every input parameter combination of every tested tool, where Q_i denotes the *i*-th sequence in *QS*:

1. Align each query Q_i against *refA* independently, obtaining alignments A_i.
2. Perform the necessary steps to place the query sequences *QS* to the *refT*. These steps may vary depending on the tested tool. To place Q_i into *refT* using alignment-based tools, A_i is used. The result of this step is a collection of placements of *QS*.
3. For every Q_i, take the placement branch $P(Q_i)$ with the highest value of likelihood reported by the tool. Create an extended tree T_i by modifying *refT* as follows. Create a new node in T_i by splitting branch $P(Q_i)$ in two branches. Attach to this new node a new terminal branch leading to a leaf labelled by Q_i.
4. Reoptimize branch lengths and calculate the LogLikelihood (LL_i) of T_i:
   `> raxml-ng --evaluate --msa A_i --tree T_i --model MODEL --redo`
   Use the MODEL parameter given by the user in a config file.

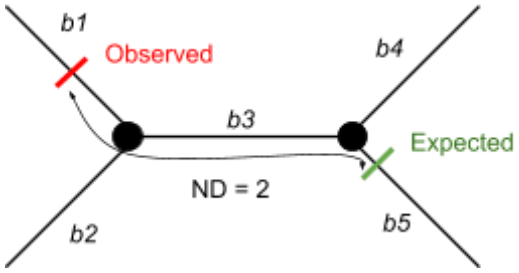
In the end, the vector containing all the LL_i values can be used to compare the performance of different PP tools and/or their input parameter combinations.

For example, if LL_i(EPAng) and LL_i(RAPPAS) denote the values obtained while using EPAng and RAPPAS, respectively, a user can inspect the distribution (via histograms/boxplots etc.) of LL_i(EPAng) - LL_i(RAPPAS).

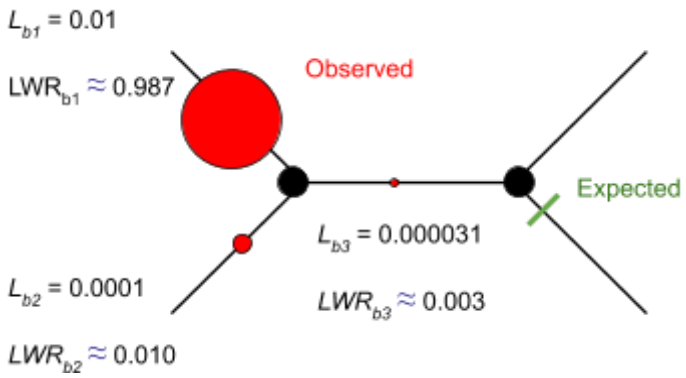
Comments on the ND and eND metrics

The ND (Node Distance) and eND (expected Node Distance) metrics were originally described in detail in the original papers of PPlacer (Matsen *et al.*, 2010) and EPA (Matsen *et al.*, 2010; Berger *et al.*, 2011). Below is a rapid description of their difference.

Difference between ND and eND metrics (PAC procedure):
Both metrics are topological measures which report, for each placed query, the number of tree nodes that separate an *observed* placement (e.g. the branch associated to the best likelihood, which is the best placement) and an *expected* placement (defined as the branch from which taxa were pruned by the pruning procedure). For instance, consider this simple tree of 2 internal nodes (black dots) and 5 branches labelled *b1* to *b5* . The ND between *observed* and *expected* placement is 2 (whatever the position of the placements along branches *b1* and *b5*) :



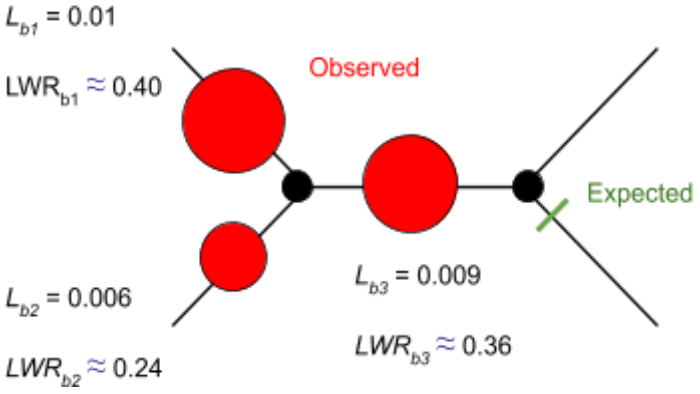
For each placement, likelihoods are computed for more than one branch of the tree. In general, phylogenetic placement tools report not only the branch of best likelihood, but the *n* branches associated to the top *n* best likelihoods. Thus, a "placement" can be seen as a distribution of likelihoods observed in one more than 1 branch. A statistic called the Likelihood Weight Ratios (LWR) is associated with each branch to take into account the relative difference observed between these likelihoods and can be seen as a measure of uncertainty of the placement.
For instance, considering *n*=3 (e.g. likelihoods and corresponding LWR are output for the top 3 best likelihoods) we observe the likelihoods and LWR, with L_{b1} being the likelihood of a placement on branch *b1* and LWR_{b1} the corresponding LWR ratio (red circles illustrate the LWR difference) :



In this particular example, it seems that *b1* is by far the best placement, and the ND and the eND are relatively equivalent (2 nodes separate *observed* and *expected* placements). The corresponding eND of this placement is :

$$eND = (ND_{b1} \times LWR_{b1} + ND_{b2} \times LWR_{b2} + ND_{b3} \times LWR_{b3}) / (LWR_{b1} + LWR_{b2} + LWR_{b3})$$
$$eND = (2 \times 0.987 + 2 \times 0.01 + 1 \times 0.003) / (0.987 + 0.01 + 0.003) = 1.997$$

Now consider this alternative situation:



The likelihoods associated with *b1* and *b3* are relatively similar, which is reflected in their LWR values. Said otherwise, while *b1* was chosen is the best placement, it appears *b3* remains a decent branch for placement. However, choosing branch *b1* or *b3* would result in different ND values (2 and 1 respectively). The usefulness of the eND measure lies in its ability to take into account this uncertainty by weighting the NDs by their associated LWR. For this second example:

$$eND = (2 \times 0.4 + 2 \times 0.24 + 1 \times 0.36) / (0.4 + 0.24 + 0.36) = 1.64$$

Recommendations for using the ND and eND metrics (PAC procedure):

A first intuition would be that the eND is a better accuracy measure than the ND as it considers the placement uncertainty represented by relatively similar likelihoods associated to different branches (see previous paragraph). However a few remarks must be made.

Considering the state of current placement tools (June 2020), we would make the following recommendations:

- APPLES can output only one branch per placement (the one associated with the best score) and LWR values are consequently always equal to 1 for the single placement branch. Until APPLES allows output more branches per placement, using the ND metric would be more fair in experiments targeting tool comparisons.
- When comparisons involve other software (EPA-ng, PPlacer, RAPPAS and not APPLES), using the eND is applicable, because all these tools can output several branches per placement (and corresponding LWR are different).
- When comparing tools, it is recommended that they output the same amount of branches per placement. In practise, EPA-ng, PPlacer and RAPPAS command-lines already share the same default output configuration (maximum 7 branches per placement and only those associated to a LWR > 0.01).

Currently, writing more recommendations on the usage of these metrics is difficult as these measures have been developed specifically for the first manuscript of phylogenetic placement and, so far, were exploited in a limited number of manuscripts and on limited number of datasets (moreover, the same datasets are used in these manuscripts).

Comments on results of Figure 1-B and 1-C

Figure 1B: Example of plots produced by the PAC procedure

This figure can be reproduced by following the example 2 of the PEWO wiki: https://github.com/phylo42/PEWO/wiki/IV.-Tutorials-and-results-interpretation#example2 The details about the configuration of the pipeline is already detailed in this online tutorial and the present section will focus on the interpretation of the results.

Warning: note that this plot was generated from a toy example of the PAC procedure limited to 10 prunings (for fast tutorials). This configuration is not necessarily representative of the actual accuracy of the tools. A better approach would be to configure the PAC procedure to test at least a 100 different prunings, which would ensure to compute both easy (a single leaf is pruned) and hard (a large subtree is pruned) simulations.

In this example, a phylogenetic tree of bacterial 16S rRNA is used as a reference tree. The goal of running the PAC procedure of PEWO on this dataset is:

1. To determine which placement software produces, on average, the most accurate phylogenetic placements.
2. For a particular tool, which parameters are optimal.

PPlacer (an alignment-based approach) is compared to RAPPAS (an alignment-free approach) and for sets of 9 (PPlacer) and 6 (RAPPAS) parameter combinations. See PEWO wiki for a more detailed explanation about the selected parameters. Accuracy is evaluated via the *expected Node Distance* metric (eND). As a reminder, the lower the eND is, the more accurate are the placements in the selected conditions.

Using the plots output by PEWO (Figure 3B), we can observe that:

- For both methods, measured eNDs are in [2,3], showing that, on average, queries are placed on a branch which is 2 nodes away from their expected placements. Considering that the corresponding reference tree shows very short branches between sister leaves, this is considered as a good accuracy. As a comparison, observe figure 3 of (Linard *et al.*, 2019) where the measured average NDs are generally above 2, whatever the reference tree considered (eNDs were not implemented at that time).
- For PPlacer, changing the parameters *ms* and *sb* (*max-strikes* and *strike-box* respectively, see Matsen et al, 2010) has a limited impact on placement accuracy, with a maximum eND difference of 0.17 between the tested combinations.
- At the opposite, RAPPAS accuracy is heavily influenced by its parameter *k* (the k-mer size) and less by the second tested parameter *o* (*omega*, which determines the amount of k-mers filtered during database construction).
- When comparing these methods, it appears that RAPPAS requires a k-mer size > 6 to be at least as accurate (k=7) or more accurate (k=8) than PPlacer on this particular dataset.

- While not represented in the figure itself but measurable with the RES procedure, one would observe that the most accurate configurations for both tools correspond to longer computations. In this regard, it appears that playing with the parameters of PPlacer can greatly accelerate the placements, while limiting the loss of accuracy. On the other hand, RAPPAS is orders of magnitude faster than PPlacer in its placement phase but will involve heavier computations when longer k-mer are used at database construction.

Figure 1C: Comparing different genetic markers

This figure can be reproduced by following the example 4 of the PEWO wiki:
https://github.com/phylo42/PEWO/wiki/IV.-Tutorials-and-results-interpretation#example4
The details about the configuration of the pipeline is already detailed in this online tutorial and the present section will focus on the interpretation of the results.

Warning: note that this plot was generated from a toy example of the PAC procedure limited to 10 prunings (for fast tutorials). This configuration is not necessarily representative of the actual accuracy of the tools. A better approach would be to configure the PAC procedure to test at least a 100 different prunings, which would ensure to compute both easy (a single leaf is pruned) and hard (a large subtree is pruned) simulations.

This example describes a possible application of PEWO procedure that goes further than the benchmarking of the placement tools themselves. In applications such as metabarcoding or metagenomics, one often has to evaluate which genetic marker is the most adapted to species identification in a sample representing a complex environmental community. In particular, one could test if different mitochondrial markers (different regions of the mitogenome) will produce more accurate species identification when considering a particular reference tree. Several PEWO runs, one for the phylogenetic tree built from each marker, can be run to answer this question and help early decisions related to experimental design.
In this particular example, four phylogenetic trees were built for four different regions of the same 1000 Coleopteran mitochondria (e.g. each tree is composed by 1000 sequences of the same species, data from Linard *et al.*, 2018). These regions are *cox1* (full CDS), *cytb* (full CDS), *12S rRNA* (full ORF) and *16S rRNA* (V2-V3 + V3-V4 regions). By using PEWO, we aim to answer the following question: using these particular reference trees, which marker is likely to produce the most accurate placements, e.g. species identifications ?
Note that the answer that will be produced with PEWO is specific to the present reference trees. If one builds a new dataset with more species or a different taxonomic composition (e.g. a phylogenetic tree with different topology and branch lengths), one should run this procedure again (different markers and tools may behave differently at different taxonomic scales).
A run of the PAC procedure is launched for each of the four different reference trees and configured to test three placement tools. It is also configured to test EPA-ng and PPlacer with

default parameters and RAPPAS with *k*=8 and *k*=10 (many more parameters conditions could be tested). The results are reported in Figure 3-C (as a reminder, lowest average eND = best accuracy). They lead to the following results and discussions:

- When considering all tools and markers, the *12S* reference tree leads to the most accurate placements when using PPlacer (eND=4.52). EPA-ng shows a very similar accuracy for this marker and RAPPAS shows a lower accuracy for both tested k-mer lengths. Still, considering that all trees are built from 1000 species and that whatever the tools and locus observed eND are inferior to 12, all methods can be considered as relatively accurate (comparatively, similar eND values measured on a tree of only 100 species would have been a worse accuracy).
- For *cox1* and *cytb,* EPA-ng and RAPPAS produce the most accurate placements, respectively. This shows that the most appropriate tool may depend on the marker. If these alternative markers are selected for the experiments, then the results suggest to use a different tool than for the *12S* reference tree (see previous point).
- For this particular set of *Coleopteran* species, average placement accuracy decreases, from *12S*, *16S*, *cox1* to *cytb*. This shows that the longest marker is not the most resolutive when using this particular reference. In fact, this particular mitochondrial reference dataset contains a large proportion of sequences belonging to the same family (*Curculionidae*). rRNA markers are known for their faster rate of evolution which consequently, and despite their shortest length, make them more resolutive for species identification at this (relatively) low taxonomic depth.
- If a metabarcoding approach is envisioned, these results suggest to build an experimental design based on the 12S marker, particularly if communities rich in Curculionidae family members are targeted in the project, and if the present (incomplete) reference database will be the basis for future species identification based on phylogenetic placements. Note that this recommendation does not necessarily hold for a different reference dataset (for instance, another reference of more even Coleopteran family sampling may conclude to the recommendation of different marker/tool/parameters).

Altogether these comparisons emphasized the usefulness of a benchmarking framework like PEWO.

References

Berger,S.A. *et al.* (2011) Performance, Accuracy, and Web Server for Evolutionary Placement of Short Sequence Reads under Maximum Likelihood. *Systematic Biology*, **60**, 291–302.

Linard,B. *et al.* (2019) Rapid alignment-free phylogenetic identification of metagenomic sequences. *Bioinformatics*, **35**, 3303–3312.

Linard,B. *et al.* (2018) The contribution of mitochondrial metagenomics to large-scale data mining and phylogenetic analysis of Coleoptera. *Mol. Phylogenet. Evol.*, **128**, 1–11.

Matsen,F.A. *et al.* (2010) pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, **11**, 538.